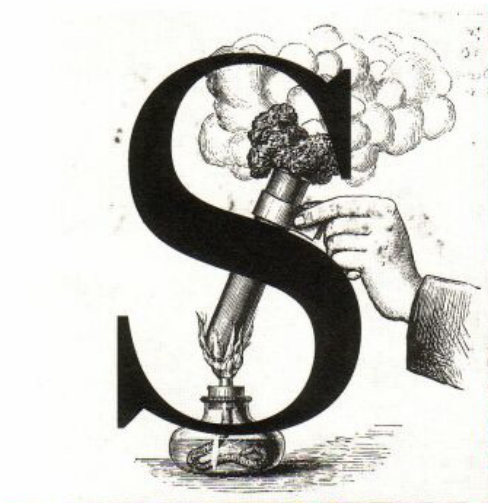
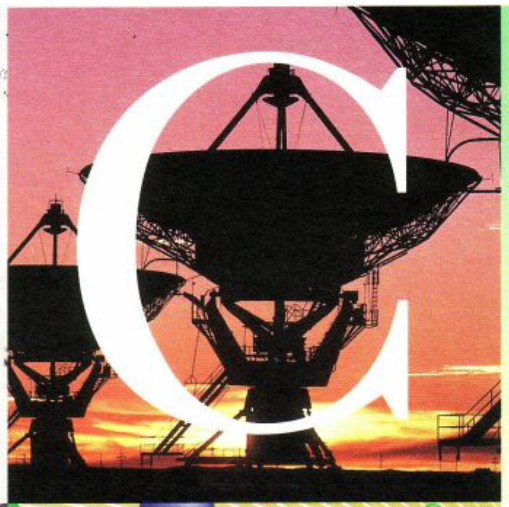
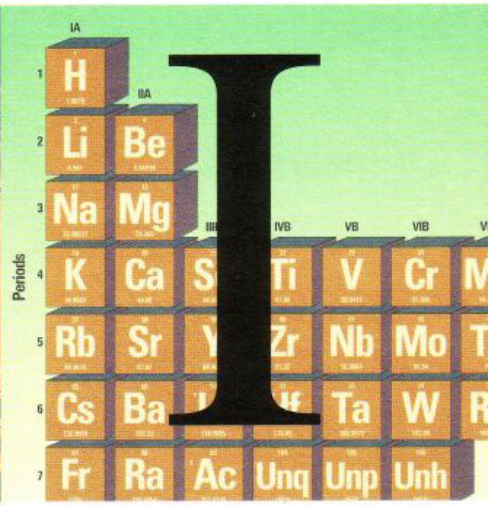
NeXT on Campus™

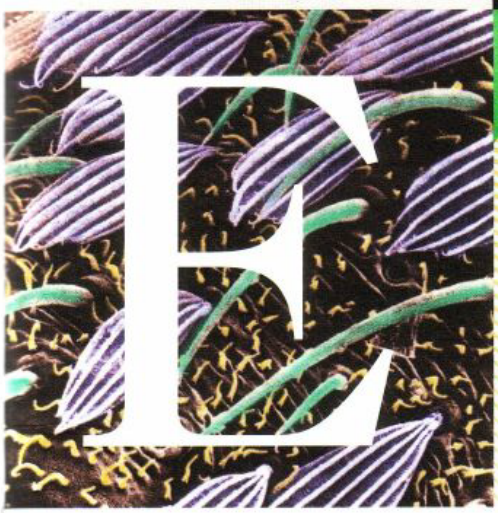
Fall 1991

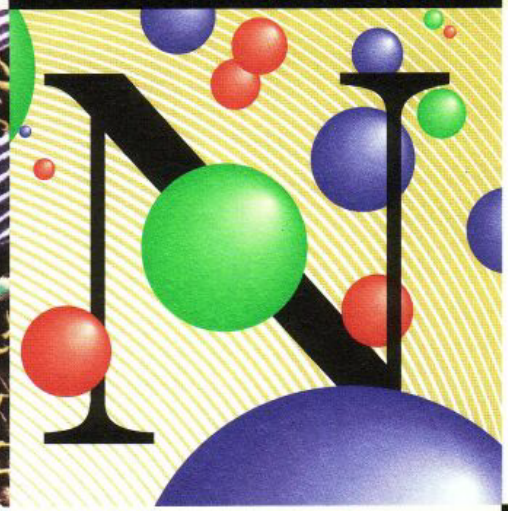
In this issue:



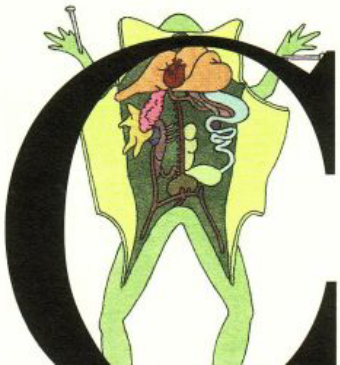








From research to publication: increasing scientific productivity with NeXT computers



Fall 1991, Volume 3, Issue 1

NeXT on Campus

Table of Contents

An introduction to extensible programming

Jeff Martin
Developer Support Engineer
Joshua Doenias
Software Engineer
NeXT Computer, Inc.

We showed our new prototype paint program to a friend recently. He made the usual comment, "It would be really great if only it had _____" (fill in the blank with your feature of choice, you've heard it before). One way to avoid having to solve everybody's requests with one program is to make your program extensible—that is, write the program so you can add features without rewriting the source code.

For example, a CD player is an extensible environment. You don't buy a stereo with ten albums built in; you buy a CD player and some CDs to go with it. When a new CD is released, you don't buy a new CD player to listen to it. And one type of CD player suffices whether you listen to classical, country, or rock 'n' roll. It's extensible!

Loadable objects—CDs for your software

Extensible applications are not new, but with the NeXTstep object-oriented development environment, creating them has never been easier. NeXTstep applications are collections of independent objects that know how to carry out functions and communicate with other objects. Window objects, for example, know how to draw themselves and move and resize themselves in response to mouse clicks. Extensible NeXTstep applications can load additional objects while they are running. Developers can write new objects that can be loaded into completed applications written by other developers. You can load these objects into your applications to give them new capabilities—just as you would add new CDs to your collection to hear new music.

Loadable objects add flexibility to NeXT applications

Applications that can load objects have unlimited flexibility. Several NeXT applications use loadable objects to increase their functionality. Interface Builder lets users load custom palettes into its palettes window. The custom palettes are then used to create advanced applications. For example, Objective Technologies, Inc. sells MathPalette,™ which can be used to create custom front ends to *Mathematica*, a powerful mathematical symbolic manipulation program.

Lotus® Improv,™ a revolutionary spreadsheet and data analysis application, also has the ability to load objects. OTProvide,™ a collection of loadable objects from Objective Technologies, lets users extract data from databases and information services from within Improv. OTProvide then creates Improv worksheets from the retrieved data. Recently, the Los Angeles County Sheriff's Office used loadable objects to turn Improv into a donation tracking, tallying, and display system for its annual S.A.N.E. Kids Say No Telethon.

Applications and environments that never go out of style

Applications that take advantage of loadable objects are more than just applications—they become application environments. Just as the Workspace Manager is an environment for running applications and managing and accessing files, Interface Builder is an environment for building applications, and Improv is an environment for analyzing and manipulating data.

Applications that use loadable objects provide users with unlimited flexibility and a customizable environment, and because new loadable objects can replace old ones, they never become obsolete. Take a circuit design tool where each chip is an object. When new chips become available, users simply load them into their application. In summary, programming with extensibility in mind lets your application satisfy the needs of different users and turns an application into an environment for research and experimentation, thus allowing it to grow long after the original development ends.

A practical introduction

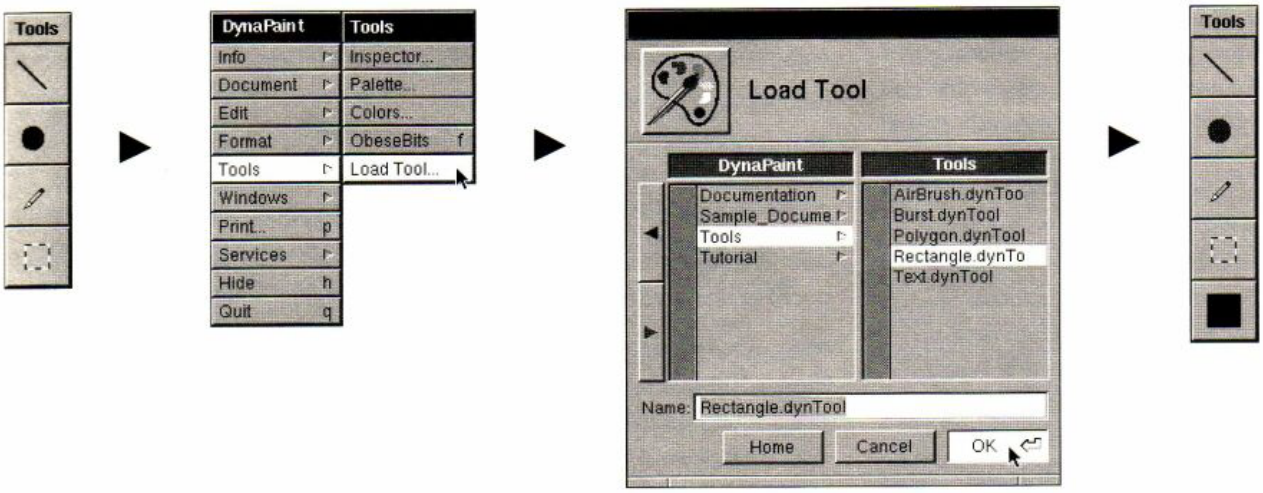
A list of programs that could take advantage of a flexible, extensible framework is long and varied. To show how this framework can be used, we'll discuss our paint program project.

As part of our development endeavors for NeXT computers, we are working on "Project DynaPaint," an extensible paint program now in the research phase of its life. It is presented here to show how an extensible program can be structured.

DynaPaint is designed to load two kinds of objects: simple painting tools and filter objects. The standard painting tools that come with DynaPaint are loadable objects. These tools include a pencil, paintbrush, rectangle, line, among others.

To create a tool, programmers write an object that responds to the following methods:

```
- mouseDown:(NXEvent *)event;
- mouseDragged:(NXEvent *)event;
- mouseUp:(NXEvent *)event;
```



Here, the Rectangle tool is loaded at run time by choosing the Load Tool menu item.

The application calls these methods in response to user actions such as a mouse click, drag, and release. The `mouseDown` method of a simple "rubberband" line tool, for example, saves the point where the user first clicks. The `mouseDragged:` method then draws a line from the original point to wherever the user drags the mouse.

Because the main program handles tasks such as redrawing, erasing, and undoing, the tools only need to know how to draw themselves. As a result, writing tools is simple and straightforward. A typical drawing tool takes up no more than one page of code.

Filter objects, which alter the selected region on the drawing canvas, are even more straightforward. The filter object must respond to only one method:

```
- filter:(NXBitmapImageRep *)
thebits
```

`thebits` is a bitmap of the selected region on the canvas. When this method is called, the filter can examine and alter the contents of the bitmap to perform image manipulation operations such as blur and contrast-adjust.

All these objects can take advantage of Interface Builder to create user interfaces. For example, when users select the Line tool, they may want to adjust the line thickness or color. Using Interface Builder, developers can create a Line Inspector window. The Inspector will get loaded into the program when needed.

On disk, tools are stored as file packages (folders that appear as a single file) and contain the following items:

- An object file that contains the code for the particular tool object
- A .nib file, created with Interface Builder, containing the Inspector window
- A text file for use in the help system

When DynaPaint loads a tool, it checks the file package for the tool object file and then uses a simple Objective C library call to load the tool. Then it loads the inspector window, and finally, it adds an icon to its tool palette to represent the

The NeXT Application Kit and the Objective C run time environment provide all you need to create extensible applications. All the tough work is done by an Objective C library function: `objc_loadModules()`; and an Application object method: `- loadNibFile:owner:-withNames:`

The `objc_loadModules()` function dynamically loads a set of Objective C objects. The program can then use the loaded objects just as if they had been linked into the application at compile time. No special code is needed.

The power to load Objective C objects at run time should not be overlooked. NeXTstep makes it easy to customize and improve applications. As more developers release programs that can take advantage of this feature, users will not only customize programs to their liking, but also trade and perhaps even sell the modules they have created.

Configuring NeXT laboratories: Achieving the optimal environment

Eric A. Larson
Technical Support Engineer
NeXT Computer, Inc.

Computer laboratories are collections of networked computers that provide faculty and students access to powerful computer equipment and network facilities. Colleges, universities, and government institutions worldwide have labs of NeXT computers dedicated to research and instruction in mathematics, computer science, music, psychology, journalism, English, economics, electrical engineering, multimedia, and much more.

Many institutions have public access or general purpose labs that are used by students and faculty for research and instruction in many disciplines. This article provides some reasons to use NeXT computers in labs, and some guidelines for choosing the appropriate hardware for your lab. NeXT computers are especially well suited for laboratories for several reasons:

- NeXT computers are powerful and inexpensive, so institutions get more computing power from smaller budgets.
- The NeXTstep development environment lets faculty and staff develop instructional and research applications that take advantage of workstation power, and are easy enough for beginning students to use.
- NeXT computers come bundled with a multitude of software. *Mathematica*, for example, the perfect tool for research and instruction in the mathematical and physical sciences, is free with every NeXT computer purchased in education. And, NeXT computers are the only workstations that run the *Mathematica* Notebook front end.
- NeXT computers support software standards including Berkeley 4.3 UNIX, MS-DOS, and X Windows. Network standards supported include software and thin and twisted-pair Ethernet, TCP/IP, and NFS. Third-party products let NeXT computers connect to IBM 3270 mainframes, AppleTalk® networks, and Novell® networks.

Here are examples of NeXT labs:

- The School of Journalism at Indiana University has over 50 NeXT computers for journalism instruction. (See "Journalism school adopts NeXT computers" on page 8.)
- University of Iowa has a lab of NeXT computers for mathematics instruction. (See "First-year calculus meets the real world," *NeXT on Campus*, fall 1991, page 8.)
- University of Maryland has 4 general access labs of 25 NeXT computers each.
- California State University at Los Angeles has 25 NeXT computers to implement its "Interactive Electronic Blackboard for Natural Science and Mathematics Education," funded by the National Science Foundation. (See "Innovative methods of education for nontraditional students" on page 12.)
- The Center for Computer Research in Musics and Acoustics (CCRMA) at Stanford University has more than 15 NeXT computers for research in computer music.

Determining lab hardware needs

Planning the hardware for your lab depends largely on its purpose. A *Mathematica* lab dedicated to solving complex equations requires more powerful hardware than a writing lab. In general, labs need at least one computer dedicated to performing various network-related tasks such as print queuing, archiving files, and keeping track of laboratory-wide computer accounts, as well as several machines to be used for actual work. Computers that perform network and administration tasks are called "servers." The remaining computers are called "clients." Recommended memory and hard drive configurations for servers and clients depend on lab size and use, but here are some guidelines.

Server configurations

Every lab needs at least one server to handle network administration. There are several additional servers you may want in your lab. This section explains the purpose of each type of server, recommends minimum system configurations, and explains reasons to use more than the minimum configurations. (In many cases, one computer can act as more than one server. One computer, for example, can be both a NetInfo™ master server and a file server.)

NetInfo master servers: run NetInfo, NeXT's network administration software.
Can be combined with: File servers
Minimum configuration: NeXTcube, 16 MB RAM, 660 MB hard disk.
Add additional memory if: Lab use is high; network traffic is high.
Add additional hard disk space if: NetInfo master is combined with a file server.

NetInfo clone servers: back up NetInfo master servers. If a NetInfo master server shuts off, the NetInfo clone server takes over the network administration tasks to ensure uninterrupted operation.
Can be combined with: File servers

What is NeXTedge?

NeXTedge℠ is a division of NeXT Computer, Inc. dedicated to network and system administration training, network design and implementation consultation, and technical support. To contact NeXTedge, call 1-800-848-NeXT.

Minimum configuration: NeXTstation, 16 MB RAM, 105 MB hard drive.
Add additional hard disk space if: NetInfo clone is combined with a file server.

Print servers: consist of a computer and a printer dedicated to serving the printing needs of a lab. Printing can tax both the network and the print server, so it should have plenty of memory.
Minimum configuration: NeXTstation, 8 MB RAM, 200 MB hard drive

File servers: provide storage space for clients. The Network File System (NFS) software included with NeXT computers lets you share folders across a network. You can use clients with small hard drives by consolidating storage space on one or more file servers, each with large hard drives. The most common types of file servers are home directory servers and archiving servers.

Home directory servers: where users store files. These servers can require significant memory. Generally, 5 MB per user is enough, though some users need as much as 25 MB. Large labs may need more than one home directory server.
Can be combined with: NetInfo master servers or NetInfo clone servers.
Minimum configuration: NeXTcube, 32 MB RAM, 660 MB hard drive.
Add additional memory if: Users frequently access large files in their home directories.
Add additional hard drive space if: There are many users requiring substantial storage

Archiving servers: act as a repositories for files that can be accessed by all lab users. Applications purchased for lab use and reference databases, for example, often reside on archiving servers. Determining hard drive size depends on how many files and applications you want to make available to the entire lab.
Can be combined with: NetInfo master servers or NetInfo clone servers
Minimum configuration: NeXTstation, 8 MB RAM, 400 MB hard drive
Add additional memory if: Users frequently access large files stored on the archiving server

Client Configurations

Since users can store files on file servers and print to print servers, clients do not need much memory or hard disk space.

Minimum configuration: NeXTstation, 8 MB RAM, 105 MB hard drive; NeXTstation Color, 12 MB RAM, 105 MB hard drive
Add additional memory if: Users often run memory intensive applications such as *Mathematica.*
Add additional hard drive space if: Users often run memory intensive applications such as *Mathematica;* if users want to store files on the client's hard drive instead of the server.

Net booting: just say no!

Some workstation vendors sell low-cost computers that have no hard drives. These computers boot from a network rather than a hard drive and rely on remote computers for all file storage and virtual memory swap space. As a result, diskless machines are slower than machines with local hard disks and slow down the network for all users. The only advantage of diskless workstations is price.

NeXT computers also can boot from a network. However, all NeXT computers come with hard drives, network, except if, for instance, the local hard drive becomes damaged. There are many reasons to boot only from the local hard drive: the hard drive provides superior performance; network traffic is reduced; if a server is out of service, clients still function; and machines that boot locally require less administration than ones that boot from the network.

Remotely mounting Software Release 2 Extended

NeXT Software Release 2 Extended requires at least 330 MB of hard disk space. Clients with 105 MB or 200 MB hard drives come with Software Release 2, the standard software release. (For information on Software Release 2, see "Upgrading to Software Release 2 and the 68040 Upgrade Board," *NeXT on Campus*, spring 1991.) To access the Extended release, you can use NFS to mount the additional Extended release files from a server onto the clients. NeXT Technical Support created a UNIX command called NfsExtend that automatically does this. NfsExtend is available to all authorized NeXT Support Centers.

To learn more

With some planning, a NeXT lab provides a powerful, productive computing environment that is easy to administer. The *Network and System Administration* manual is a good reference for learning to plan and maintain your NeXT lab. See especially, Chapter 1, "Using This Manual to Plan Your Network." Via the Internet, there is a NeXT lab mailing list you can subscribe to. Send a message to next-lab-request@cs.ubc.ca to join the list.

In focus: Diagram!
Creating and using presentation graphics

Diagram!™ created by Lighthouse Design, Ltd., is a full-featured object-oriented drawing application tailored for people who use and create drawings and diagrams in their work. Unlike traditional drawing programs, Diagram! is optimized for ease of use, allows existing documents to be revised quickly, and offers unparalleled integration with other applications. Diagram! is available with academic prices, making it an ideal choice for both faculty and students.

Palettes make creating drawings quick and easy

People who use graphics programs tend to spend more time creating versions of the same type of drawing rather than creating new drawings. For example, you might draw a series of similarly formatted charts for a report or use a standard format for all your presentations. Diagram! speeds the process of creating similarly formatted drawings: rather than choosing a tool from a tools menu and using it to draw a graphic object, you drag frequently-used graphics objects off a palette and into your drawing. Dragging graphics off a palette is quickly repeatable, whereas switching between tools from a tools menu and drawing different shapes can be clumsy and time consuming.

Custom palettes: personalizing your drawing environment

You can create your own Diagram! palettes by drawing objects and pasting them onto a palette. By creating custom palettes, you can customize Diagram! to draw the kind of drawings you use most often.

Consider an instructor who is planning a series of charts that consist of boxes, titles, and text. He might decide that the boxes should be light blue, the titles 18-point Times® Roman, and the body text 14-point Helvetica® With a traditional graphics program, each time the instructor needs a box, he must select a tool, draw a box of the proper size, and then adjust its attributes to make it the proper color.

With Diagram!, to create a custom palette, the instructor draws one box and makes it blue; draws one sample title, makes it the proper font size and type face; and draws one sample block of text, setting the proper font size and typeface. He copies these objects and pastes them onto a palette. When constructing the charts, each time he needs a box, he drags one from the palette and places it in his document. Because the box on the palette is already formatted, the box he drags into his document is already formatted with the proper size and color.

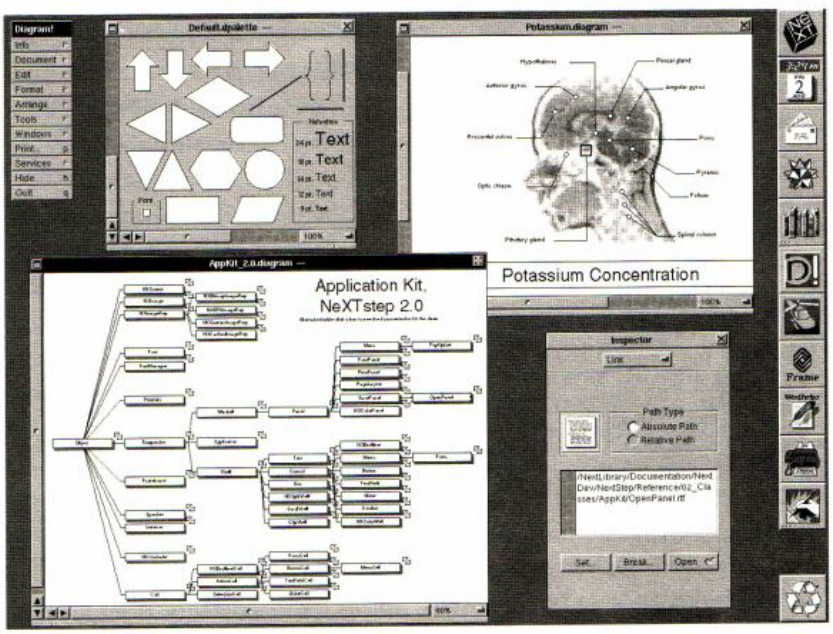
Custom palettes can be saved and reused or given to other Diagram! users. For instance, the instructor might save his chart palette and give it to other department members to standardize all their charts.

Rubberbanding lines make revision a single-step process

Revising previously created drawings can be one of the most time-consuming aspects of creating a drawing. The revision process includes moving and adjusting lines, connections, and labels. Traditional graphics packages treat lines as plain graphics, disregarding their meaning as connections. As a result, if you draw lines pointing from labels to parts of your drawing and then decide to move the label or the drawing, you have to carefully redraw all the lines.

Diagram! eliminates the need to redraw lines by letting you anchor lines to graphics objects. In a flow chart, for example, if you move a

Some of Diagram!'s advanced features included rubberbanding lines and hot-links to files.



node (a shape within a chart), all the lines and labels that point to the node redraw themselves and stay connected to the node. This is called rubberbanding. It means you no longer have to redraw lines each time you change your drawings. Rubberbanding turns modification and revision of a complicated drawing into a single-step process.

Here's a more complex example using rubberbanding lines: A biochemist might need to annotate a laboratory scan generated from his university's magnetic resonance image (MRI) machine. A traditional drawing program only lets him draw labels on top of the brain graphic. With Diagram!, the lines and labels can be anchored to particular points on the image. If he moves the labels, Diagram! redraws the lines and labels to make sure they point to the same location on the brain. Similarly, the image of the brain can be moved without losing the correct placement of its lines and labels. No matter how many changes the biochemist makes to the drawings, he only has to express the relationship between the labels and the image once—the first time. From then on, Diagram! makes sure the lines and labels are redrawn properly.

Diagram! links pictures to information

With Diagram!, you can link documents to drawings. To link a document to a Diagram! file, you drag the document's icon from a Workspace Manager File Viewer and place it on a graphics object in your drawing. A link symbol appears where the document is placed. When someone double-clicks the link symbol, Diagram! will message the appropriate application to open the document. In this way, Diagram! drawings can be used to navigate through collections of documents or to present additional information not displayed in the drawing. For example, you could link a diagram of a physics experiment to the Improv models that hold the data and analysis. Or, you could link annotated three-dimensional surface plots to the *Mathematica* Notebooks that generated them.

Consider also a computer science student taking a course in object-oriented programming. He could draw a diagram detailing the hierarchy of object classes in the NeXT Application Kit. By linking the technical documentation that describes each object class to the chart, he can use the chart to quickly retrieve the specifications of any object.

Sound

Diagram! users can place recorded voice messages on graphic objects in their drawings. With voice messages, Diagram! documents can become self-running sets of instructions and presentations used without the presence of an instructor.

All this and the usual

In addition to custom palettes, automatic rubberbanding lines, document links, and sound annotation, Diagram! offers a complete set of traditional draw program features. Diagram! also has advanced drawing features, such as multiple pages, gridding, grouping, zooming, layering, alignment, and distribution.

Academic pricing

Diagram! is available to college and university faculty and staff for $75 tion version of Diagram! is the same full-featured drawing package available to commercial buyers for $399 but is shipped without packaging, support, or hard-copy documentation. Documentation is shipped on-line and is indexed for Digital Librarian. Diagram! is available through NeXTconnection℠ at 1-800-800-NeXT, or (603) 446-3383. A demonstration version of Diagram! is available free via anonymous FTP from the nova.cc.purdue.edu archive site. (For information on public archive sites, see "NeXT Archives" on page 30.)

New from Lighthouse Design

In addition to Diagram!, Lighthouse has added VOID™ to its line of applications for NeXT computers. VOID is a space-based, multiplayer video game that can be played against the computer or against other network users. Highlights include three-dimensional rendering (with an outline mode for 68030 users), interactive dogfighting, DSP-generated sound, and inter-cockpit messaging for networked games.

VOID is now shipping in three paks; each copy supports three simultaneous users. Additional 3 paks can be added to increase the number of network users to a practical limit of 18. Contact NeXTconnection for details.

A portion of Lighthouse's proceeds are donated to charity annually. Lighthouse does not use shrink-wrap or Styrofoam products, and guarantees its products for the life of the owner. Lighthouse is eager to hear from users: diagram@lighthouse.com

TELL THE WORLD HOW YOU USE NeXT COMPUTERS, AND YOU MAY GET SOME FREE NeXT SOFTWEAR

At NeXT, we're interested in helping faculty, researchers, and students, tell the world about their work with NeXT technology. When users write to us, we publish stories in *NeXT on Campus* about their work, and the news travels to thousands of NeXT users around the world.

Turn to page eight and read what Professor Wil Counts at Indiana University told us about his work in journalism. Or, find out about the latest higher energy physics research on page five from Ernest Prabhakar, a Ph.D. candidate at California Institute of Technology.

Tell us how you use NeXT computers, and if we write about you in a future issue of *NeXT on Campus*, we'll send you an official NeXT T-Shirt!

Please send us a brief description (approximately 500 words) of how you use NeXT computers. In addition, we need: your name and title, institution name and department, U.S. Mail and e-mail addresses, and daytime phone number. Send to: Academic & Research Projects
NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063
Or, submit your description via e-mail:
next_on_campus@next.com

NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063

Bulk Rate
U.S. Postage
PAID
Permit #3951
San Jose, CA